Chapter

Virtual Sensors for Smart Data Generation and Processing in AI-Driven Industrial Applications

Maddi Etxegarai, Marta Camps, Lluís Echeverria, Marc Ribalta, Francesc Bonada and Xavier Domingo

Abstract

The current digitalisation revolution demonstrates the high importance and possibilities of quality data in industrial applications. Data represent the foundation of any analytical process, establishing the fundamentals of the modern Industry 4.0 era. Data-driven processes boosted by novel Artificial Intelligence (AI) provide powerful solutions for industrial applications in anomaly detection, predictive maintenance, optimal process control and digital twins, among many others. Virtual Sensors offer a digital definition of a real Internet of Things (IoT) sensor device, providing a smart tool capable to face key issues on the critical data generation side: i) Scalability of expensive measurement devices, ii) Robustness and resilience through real-time data validation and real-time sensor replacement for continuous service, or iii) Provision of key parameters' estimation on difficult to measure situations. This chapter presents a profound introduction to Virtual Sensors, including the explanation of the methodology used in industrial data-driven projects, novel AI techniques for their implementation and real use cases in the Industry 4.0 context.

Keywords: virtual sensors, artificial intelligence, machine learning, innovative sensing strategies, internet of things, industry 4.0

1. Introduction

Digitalisation and data exploitation are two of the fundamental driving forces of the new paradigm defined by the Industry 4.0 (I4.0) revolution. Recent developments in sensors, Cyber-Physical Systems (CPS), automation, and quality inspection, among others, are motivating the digitalisation of the manufacturing and non-manufacturing industries, making available large amounts of data that may capture the nature of the process and its variability. These data streams become of utmost importance when targeting enhanced productivity, flexibility, competitiveness, and environmental impact. Hence, these large data streams not only represent a valuable opportunity but also introduce a substantial challenge for industries to digest and extract value from them, without losing focus on their day-to-day operations. Data-driven solutions, including Data Mining, Big Data, or Artificial Intelligence

(AI), provide the right tools and functionalities to digest these large amounts of data, create value, and impact manufacturing operational Key Performance Indicators (KPIs). Moreover, AI-based solutions can also support knowledge discovery actions and enrich experts' industrial knowledge by discovering previously unknown process parameter correlations that can have a big impact on industrial operations.

The perceived value of data exploitation techniques, mainly powered by AI solutions, has increased in line with the growing available data in nearly all processes and sectors. The development of data-centred and data-driven solutions has become a crucial element as a tool for not only managing but also taking advantage of the incoming process data. Nevertheless, an important issue must be considered: do available or captured data accurately represent the scenario, the process, and the environment? In most cases, the answer is no. Not all relevant or key process parameters can be physically measured, or the associated cost for direct measurement is not sustainable. Thus, the need for computing or estimating these key process parameters based on measured data has become more relevant as data availability has increased and production excellence has become progressively more demanding.

Traditionally, relying on the data provided by physical sensors has been a recurring challenge due to several limitations: the cost of the sensor, accuracy, stability, and impossibility to measure specific parameters due to physical, spatial, or environmental constraints. These challenges have commonly been addressed by using analytical approaches based on physical and mathematical expressions. While this strategy can increase the underlying physics knowledge of the process and provides a general solution, it also requires extensive experimental validation and the definition of accurate assumptions and boundary conditions. Recent AI and Machine Learning (ML) advances allow for novel data-driven approaches to estimate key process parameters. The so-called Virtual Sensors (VS), also known as Soft Sensors or Software Sensors, represent a software layer that provides indirect measurements of a process variable based on the data gathered by physical (or other virtual) sensors leveraging a fusion function [1]. The exponential growth of data during the last decade has entailed the rise of data-driven solutions powered by AI and ML algorithms, correlating input data (measurable parameters) with output targets by heuristic and probabilistic models.

This chapter aims to explain the potential of Virtual Sensors for industrial process monitoring and provide an introduction to their development. Two real industrial use cases are presented, focused on High Pressure Die Casting and wastewater treatment, to illustrate and highlight the capabilities of this technology.

2. Industrial applications of Virtual Sensors

The decision-making process in industrial applications (logistics, planning, quality control, predictive maintenance, etc.) is driven and influenced by the evolution of key parameters along the production process value chain. In most cases, this set of key parameters is obtained by deploying sensors along the process chain. For instance, placing a thermocouple sensor to monitor the temperature in a foundry or a flow meter in a complex water distribution system pipe. The monitoring data captured along the production line is used to compute KPIs that measure the operational performance of the industrial process or equipment over time. Industrial KPIs are of utmost importance for informed decision-making, as well as for measuring and targeting an objective accomplishment. Some of the most relevant KPIs are:

- **Throughput**: the number of produced units per time unit.

- **Scrap ratio**: the number of defective parts over the total production.

- **Availability**: the ratio between uptime and production downtimes.

- **Overall Equipment Effectiveness (OEE)**: The percentage of manufacturing time that is genuinely productive, combining aspects of quality, performance, and availability in a single KPI.

Data accuracy and reliability are of great importance since the decision-making process relies on KPIs computed from data gathered at the production chain. In case of sensor failure, malfunction, drift, or need for recalibration, industrial KPIs may not confidently represent the process performance anymore. This situation could lead to two non-optimal scenarios: poor decision-making due to the lack of reliable information or production breakdowns due to equipment failure. Furthermore, equipment, infrastructure, material, or even people involved (technicians, staff, etc.) may be threatened due to the malfunction of the monitoring systems. Thus, mitigation strategies should be considered to reduce this risk. Robust and accurate data-driven solutions leveraging production data can provide resilience capabilities to operate in non-optimal conditions. Virtual Sensor offers an appropriate solution since they increase the reliability and agility of the system at a low operational cost, providing an indirect measurement for non-measurable physically properties.

AI and data-driven Virtual Sensor can significantly impact industrial applications by providing valuable process insights that support and enrich informed decision-making processes, as shown in **Figure 1**, where the schematic design of two Virtual Sensors is introduced.

Within the industrial applications, the three key objectives of Virtual Sensors are:

- **Expand knowledge:** To compute extra parameters derived from real sensors that are impossible or not sustainable to measure (at full-scale), thus contributing to a better understanding of the process.

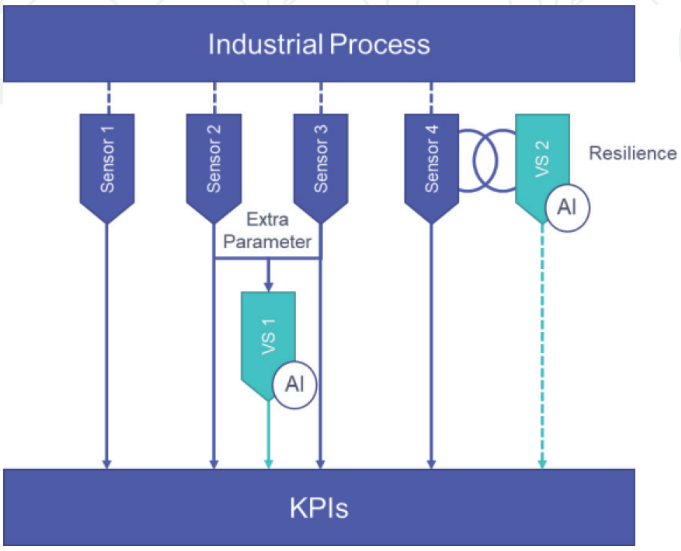

Figure 1.
Virtual sensors applications in industry.

- **Resilience:** To simulate real sensor outputs that mitigate production breakdowns due to equipment failure or even planned maintenance.

- **Accuracy:** To remove and replace the occurrence of outliers in real sensor readings and detect sensors drifts and recalibration needs.

Industrial applications can benefit from the Virtual Sensor functionalities: increasing the knowledge of the process, reducing the operational costs of the monitoring strategy, and offering a cost-effective solution enhancing monitoring system resilience.

Even though Virtual Sensors are a relatively recent research topic, their industrial applications are becoming increasingly relevant. A promising example is the usage of Virtual Sensor in Smart Factories and digitalised manufacturing facilities where devices, machinery and production systems are interconnected to enhance decision-making and management [2]. Dobrescu et al. [3] presented the development of services and computing resources for hybrid simulation of Virtual Manufacturing systems, providing a sensor-cloud interface where the end-user can virtualise multiple Virtual Sensors. The adoption of robots and their interactions with humans in Smart Factories was studied by Indri et al. [4], where Virtual Sensors were used to enhance the knowledge of the robot operation.

The applications of Virtual Sensor in the manufacturing industry are very heterogeneous. Maschler et al. [5] estimated the combustion duration on a large gas engine using just the rotational speed as input data. They studied in this work the importance of pre-processing the data for greater accuracy, showing different results for the use of Principal Component Analysis, Fast Fourier Transformation, or just a simple smoothing of the measured rotational speed. Alonso et al. [6] aimed to calculate the cooling power estimation to enable the replacement of the expensive portable measuring system. They used a model based on a Deep Learning architecture that involved data from the chiller's thermodynamic variables (temperature and pressure) and data from the refrigeration circuit (pressure power).

Other studies focus on the malfunctioning of the system instead. Zenisek et al. [7] presented an approach to stabilise and optimise the metal deposition process, merging information from various sources. The ML-based method generates a valid data stream from heterogeneous sources and can mitigate the problem of data merging through the knowledge of domain experts. Finally, they presented a real use case where they estimated the current weld bead height, one of the principal performance indicators of the process. Aware of the problems that could generate a sensor failure and the consequent interruption of information flow, Ilyas et al. [8] introduced a framework capable of finding Internet of Things (IoT) sensors in the surrounding environment and replacing faulty sensors in an automated way. The framework selects the data source based on metadata description, pre-processes historical data, and trains and ranks machine learning algorithms with great results without human intervention. They tested the model predicting the output of a solar power plant.

Tegen et al. [9] proposed Dynamic Intelligent Virtual Sensors (DIVS). The idea was to combine a broader (and not fixed) set of heterogeneous data sources based on Machine Learning to involve the user in the loop. The dynamic part of the concept can be interesting for industrial applications: evaluating the inputs of the Virtual Sensor in terms of information quality (for instance, noise, entropy, etc.) and deciding whether a data source (physical sensor) should be removed or added to the Virtual Sensor. Moreover, the online incremental learning concept was also applied, looking for a Virtual Sensor that relies not only on traditional batch learning but can be dynamically adjusted involving user labelling.

Virtual Sensor can also be applied in multiple areas of the industrial water domain covering the whole water cycle. Djerioui et al. [10] implemented a Virtual Sensor of the chlorine parameter in water treatment plants using the conductivity, dissolved oxygen, suspended solids, and pH variables as input data. The study compares the performance of a Support Vector Machine (SVM) and an Extreme Learning Machine (ELM) ML algorithm, showing better behaviour using ELM. Pattanayak et al. [11] developed a Virtual Sensors to predict in real-time the Chemical Oxygen Demand (COD) of the river Ganga using the input quality parameters of ammonia, total suspended solids, nitrate, pH, and dissolved oxygen. They evaluated different algorithms, finally building a predictive model based on K-Nearest Neighbours, which was used to predict the water quality at the treatment plant's discharge point.

Wastewater treatment is a process where factors such as energy cost or climate footprint are directly related to the process optimisation. Virtual Sensor enables monitoring key parameters in situations where the physical sensors may lead to error due to the constant contact with wastewater. Foschi et al. [12] proposed a Virtual Sensor for the *E. Coli* value for wastewater disinfection using the data from conventional wastewater physical and chemical indicators (such as COD, nitrate, and ammonia). Their research obtains a predictive model trained using an artificial neural network, which could save up to 57% of disinfectant. Pisa et al. [13] showed a Virtual Sensor to predict ammonium and total nitrogen to control effluent violations at the treatment plant using input flow, input ammonium, temperature, and internal recycle flow data. They accomplished the generation of a predictive model using a deep neural network with Long-Short Term Memory neurons, capable of predicting the nitrogen-derived parameters with good accuracy.

3. Methodology

In this chapter, we focus on the Machine Learning domain, currently one of the most trending areas under the Artificial Intelligence umbrella. Machine Learning aims to develop smart models based on data-driven algorithms that can accurately generate predictions without the explicit necessity to program them for that objective. It can be seen as learning (or training) a function (f) that maps input variables (X) to output variables (Y). Once defined, function f can be used to generalise the learned behaviour and make predictions (Y′) given a new unseen instance of input variables X'. Here, a data-driven approach depends on existing data sets to infer the unknown function f based on parametric or non-parametric algorithms.

More specifically, we propose the use of the regression-type of the Supervised Learning family of algorithms for the Virtual Sensor implementation, as shown in **Figure 2**. These algorithms rely on labelled datasets providing both input variables X

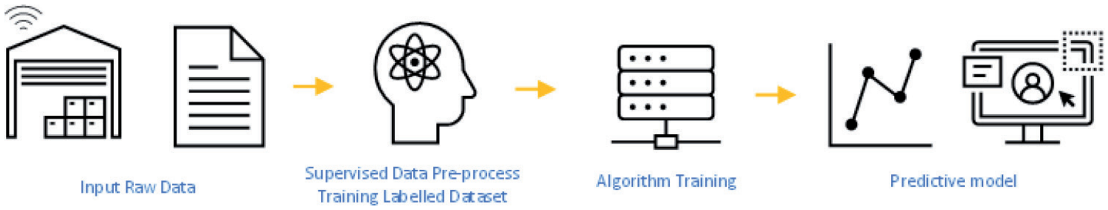

Figure 2.
Supervised learning paradigm.

and output variables Y to infer the function f. Moreover, in regression problems, the output variables Y are continuous values instead of the categorical data type required in classification problems.

In this scenario, to successfully conduct a data-driven project, it is of utmost importance to follow a standardised method to translate business problems into tasks, suggest data transformations, or provide means for evaluating the final results and reporting the process, among other objectives. The Cross Industry Standard Process for Data Mining (CRISP-DM) methodology provides this flexible framework [14] and it is organised into seven well-differentiated phases, as shown in **Figure 3**. In this sense, the data mining process is generally cyclic, since it is usually necessary to go back and forth between stages until a valid solution that meets the quality criteria is found. At this point, it is usually a common misunderstanding across the community to consider that the work is finished. Even when the solution is finally deployed and integrated into a production environment, the performance of the underlying models needs to be continuously checked. This is due to the data-driven nature of the concept, which could make a model unfeasible, for example, in those cases where the baseline conditions of the studied process change or evolve over time. This effect makes the learned function f not valid for new scenarios since the relation between input variables X and output variables Y has changed. An innovative solution in this scenario considers an online CRISP-DM model to retrain and validate the predictive models periodically over time.

The CRISP-DM process starts with understanding the business perspective, objectives, and requirements to design the project plan together with the field expert. Once the goals are defined, the initial data are collected and processed with activities channelled to familiarise with them. This first analysis can help identify data quality problems or detect interesting subsets to enable hypotheses for hidden information. Next, the data preparation phase aims to construct the final dataset, which will be used to feed and validate the algorithms. Usually, a significant amount of effort is devoted to this task since it is the most time-consuming and delicate stage that generates one of the most critical outcomes, the training dataset. This is important because data must be consistent and reliable in the Data Science domain since it defines the basis of all the solutions. Data cleaning, feature engineering, feature selection, or data scaling are some of the common processes carried out in this stage and require experienced and creative data scientists for a successful implementation.

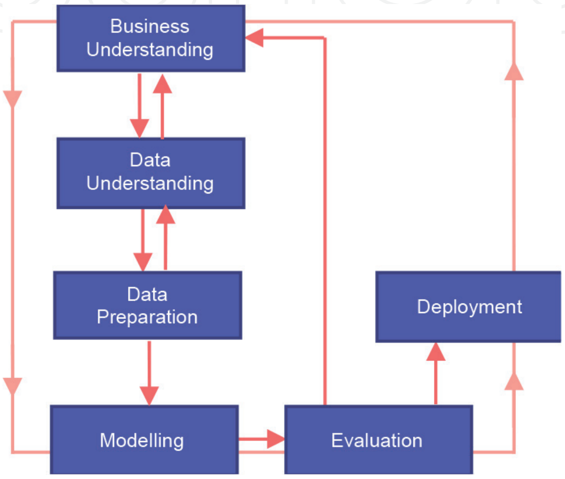

Figure 3.
Phases of the CRISP-DM process.

Different Machine Learning algorithms are selected, trained, and calibrated in the modelling phase to achieve optimal performance. The reason behind trying different algorithms is that each one is based on several techniques, has different mathematic fundamentals, and makes different assumptions. Thus, it does not exist one general solution to all the problems and each case needs to be analysed independently, given the fact that underlying data and patterns are different [15].

Then, even though the algorithms are independently evaluated in the modelling phase, during the evaluation phase, the whole model, all the stages, and all the algorithms that appear should be thoroughly assessed and reviewed, as well as the business objectives defined in the initial business understanding phase. Furthermore, a comparison across different models is required to identify the most successful ones. Finally, for the deployment, the model and the knowledge gained are organised and presented in a way that the final customer can understand, use, and maintain.

Usually, the model training, selection and evaluation stages follow a well-established methodology in the Data Science domain, as shown on the left in **Figure 4**. First, in case of parametric Machine Learning algorithms, the model training step is aimed at learning and validating the parameters of the function f (e.g., the coefficients in regression or the weights of a neural network). Separate training and testing datasets must be used across these phases. Otherwise, the model would suffer from overfitting. In this case, a model that reproduces training labels Y during the validation would present a perfect score but would not be able to make good predictions on new data X', since it would not have learned the authentic data patterns.

Splitting data into training and test datasets is known as the Cross-Validation (CV) process, and K-fold CV defines its most basic implementation. The idea is to split the training dataset into k folds, train k models using k-1 different folds (as training datasets) and validate them on the remaining fold. Several methodology variations have been proposed depending on the data type, the basic idea of this concept is shown on the right in **Figure 4**. The final training performance corresponds to the average of the k individual models' performance.

The training step also considers the evaluation and search of the most optimal algorithm hyperparameters given a training dataset. In this context, the hyperparameters are those algorithm parameters that by changing their value, are used to manage the learning process (e.g., the learning rate in Gradient Descent-based approaches). Similar to the CV procedure, several methodologies were used to this end [16]. To mention some, Grid Search proposes an exhaustive search on all hyperparameter combinations given a set of predefined values, while Randomised Search samples any given number of candidates from a parameter space following a specified distribution.

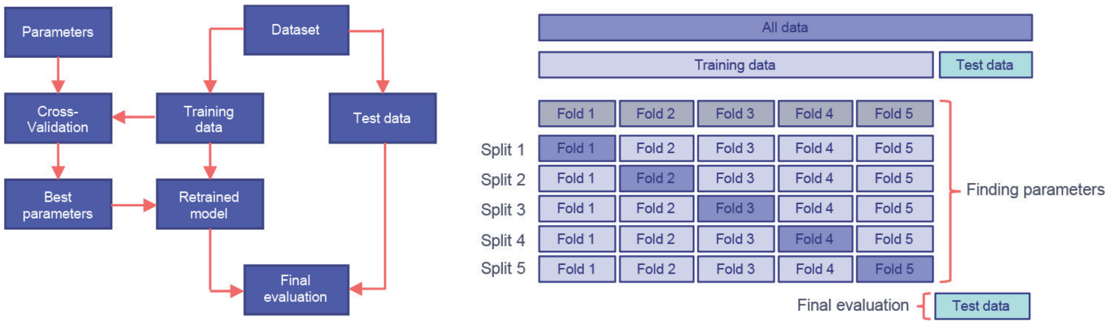

Figure 4.
Left: Cross-validation flow in ML models training. Right: K-folds CV approach.

Finally, to correctly understand the presented Virtual Sensor case study's performance, it is also essential to introduce the validation metrics used to evaluate and compare the models. The following regression metrics are proposed:

- **Mean Absolute Error (MAE) regression loss**: computes the averaged absolute difference (error) between the ground truth and the model predictions.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right| \tag{1}$$

where N is the test dataset size, y_i is the ground truth value of the i_{st} entry in the test dataset and $\hat{y}_i$ is the model prediction of the i_{st} entry in the test dataset.

- $\boldsymbol{R}^2$ **or the coefficient of determination:** computes the proportion of variance explained by the independent variables in the model.

$$R^2\left(y_i, \hat{y}_i\right) = \frac{\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{N}\left(y_i - \overline{y}\right)^2} \tag{2}$$

where N is the test dataset size, y_i is the ground truth value of the i_{st} entry in the test dataset, $\hat{y}_i$ is the model prediction of the i_{st} entry in the test dataset, and $\overline{y}$ is the average ground truth value.

4. Case studies

Virtual Sensors are a flexible and versatile technology that can be found in multiple sectors of the industry. In this section, two real use cases are introduced. The first case is related to mould injection of metallic pieces in the manufacturing industry. The second case is related to the wastewater treatment industry.

4.1 Aluminium mould injection use case

High-Pressure Die Casting (HPDC) is a process in which a molten metallic alloy is forced under pressure into a locked metal mould cavity, formed by the cover die half and the ejector die half, where a powerful press holds it until the metal solidifies. After solidifying, the ejector die half opens, and the piece is ejected. Finally, the dies are closed again, ready for the next cycle. The casting process is composed of 3 stages:

- Prefill or slow shot stage: the plunger advances at low speed until the metal starts to fill the dies cavity.

- Fill or quick shot stage: once the metal reaches the gate of the die, the plunger speed is sharply increased, between 4 and 10 times.

- Consolidation or solidification stage: once the dies cavity is filled with about 95–98% of its volume, the plunger reduces its speed, and the controlling variable is switched from plunger position to pressure, inducing a high pressure during the metal solidification process.

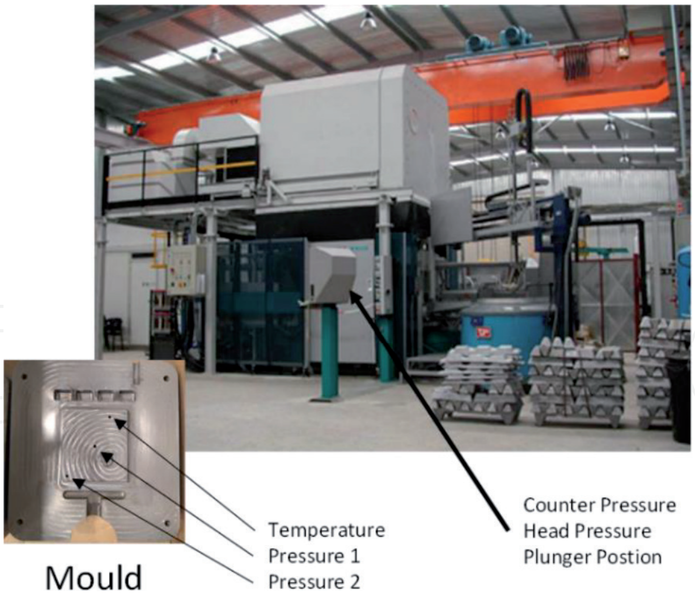

Figure 5.
HPDC Bühler machine with the three machine sensors (counter pressure, head pressure, plunger position) and an image of the mould with the holes of the three mould sensors (temperature, pressure 1 and pressure2).

The HPDC machine incorporates many sensors to track its activity. However, the mould, which must be redesigned for each new piece or batch, should include additional sensors if its sensorization is needed. As mould sensors are expensive, difficult to instal, and their integration may affect the product's finish, the proposed solution is to replace the in-mould sensors with Virtual Sensor, inferred using external machine sensors data. The Virtual Sensors allow to monitor the process and to apply corrective and preventive actions. These Virtual Sensors are developed using AI and ML methods, enabling a richer and more profound understanding of the process. The HPDC machine used for these experiments, the mould and the sensors are shown in **Figure 5**.

4.1.1 Data

The experimental campaign is carried out in the Bühler Evolution D53 machine, where aluminium L2630 is injected into tray-shaped moulds. During the lapse of two days, 256 pieces are cast at 13 different machine configurations. For each machine configuration at least 10 samples are manufactured. During each batch, the data from six sensors is recorded at a 2 kHz frequency. The change of the three in-mould sensors, two for the cavity pressure and one for the cavity temperature is shown on the left graphic in **Figure 6**. The temporal evolution of the three machine sensors: plunger position, head pressure and counter pressure is shown on the right plot in **Figure 6**.

The dataset recorded during the first day (132 tests) is used to train the model while the dataset of the second day (126 tests) is used for the test phase.

4.1.2 Methods

As previously explained, the in-mould sensors are expensive and may affect the shape and result of the final piece. Therefore, in this section, the in-mould sensors: a temperature sensor and two pressure sensors, from now on referred to as pressure 1 and pressure 2, are predicted using machine sensors: plunger position, velocity, head pressure and counter pressure. This part exemplifies the Virtual Sensor forecasting.

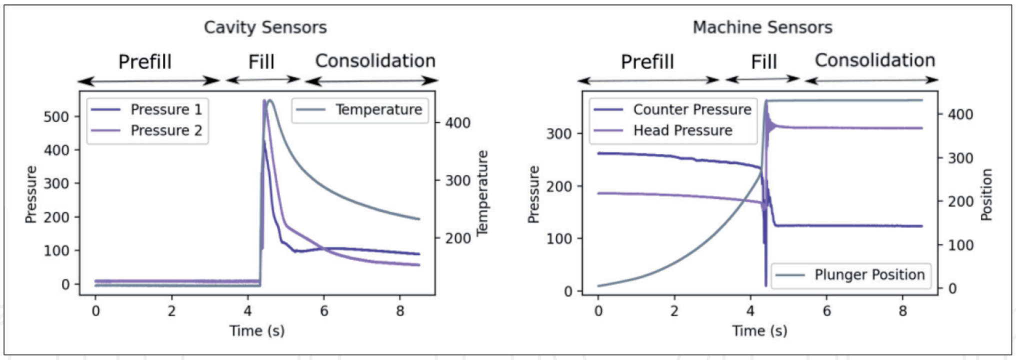

Figure 6.
Schematic representation of an HPDC shot sleeve and an injection curve with the 3 different phases: Prefill, fill and consolidation.

Following the CRISP-DM methodology, the following phase is the data preparation, essential to arrange the input data that is later fed to the algorithms. The Pearson's correlation analysis [17] demonstrates that the plunger position, counter pressure, and head pressure are highly correlated. Thus, the products of these three sensors, in pairs, are added as variables: counter pressure x head pressure, counter pressure x plunger position and head pressure x plunger position. This technique enables the use of highly related variables while preserving their influence.

To predict an instant value of any of the three virtual samples defined, the three original sensors, the three products explained above, the derivative of the position (the velocity), and 2 or 5 back samples are given as input, iterating through the results to find the most suitable input parameters for this use case.

The training dataset is split randomly in a stratified way, keeping the same percentage of machine configurations in each. The 80% of data are used for the training dataset, and the remaining 20% are employed in the validation dataset. Finally, all the data are scaled using the *MinMaxScaler,* which transforms the data into the 0–1 range. Training data are first fitted and, afterwards, train and validation datasets are converted.

The CV grid search methodology with the aforementioned K-fold split is implemented to train an evaluate different models based on the following ML algorithms:

- **Decision Tree:** an algorithm that predicts the target value by learning simple decision rules inferred from the data in a flowchart-like tree structure, with decision nodes and leaves. The chosen hyperparameters to tune are the maximum depth of the tree that will be created, and the maximum number of features required for each split [18].

- **Random Forest:** an ensemble regression algorithm that computes the output by randomly generating a multitude of decision trees and averaging the predictions of all the trees. The chosen hyperparameters to tune are, like the Decision Trees algorithm, the maximum depth, and the maximum number of features. Additionally, the number of trees in the forest (the number of estimators) has also been chosen [19].

- **KNN:** the k-Nearest Neighbour algorithm predicts the output by storing all the training data and calculating the distance between the new and stored data. The most important parameters are the number of neighbours used to predict each data point and the weight function used to determine the importance given to the neighbour data [20].

- **SVR:** Support Vector Regression algorithm is a variation of the classifier Support Vector Machine but adjusted for regression problems. Instead of separating data into classes by means of a hyperplane, the data are adjusted to the mentioned hyperplane with a certain degree of tolerance given (ε), where the best fit is the hyperplane with the maximum number of points. Therefore, the hyperparameter ε needs to be tuned, together with the C parameter, which determines de regularisation applied to the algorithm [21].

4.2 Wastewater treatment plant use case

The Activated Sludge Process (ASP) [22] is usually a critical stage in a Wastewater Treatment Plant (WWTP) and has a direct impact on the effluent water quality as well as on the greenhouse gas (GHG) emissions, demanding considerable quantities of energy. Specifically, the ASP Nitrification step is the biological process of converting ammonia to nitrate in wastewater tanks using aerobic autotrophic bacteria. The process requires proper working conditions such as enough biomass concentrations, specific environmental conditions, a minimum residence time to process the water, and a great amount of oxygen. Any variation in these conditions directly affects the amount of ammonia being treated, thus in the effluent water quality.

In this scenario, the airflow system controls the oxygen injection, one of the key processes with the highest resource consumption and impact in the treatment plant. The water operators manage the air blowers to optimise the process (i.e., the effluent water quality reaches the expected criteria, while energy consumption and GHG emissions are minimised), thus the use of sensors to monitor in real-time these quality parameters enable an online control. Ammonia is another key parameter that needs to be adequately treated. In case its monitoring gathers non-real values, the blower's management is directly influenced, resulting in elevated costs, climate impacts and issues in the effluent water quality. Implementing a Virtual Sensor enables continuous monitoring of the ammonia parameter which enables the: i) detection of sensors' malfunction or drift in measurements (due to the constant contact with wastewater), and ii) implementation of maintenance actions without the need to stop dependent systems, therefore ensuring correct and continuous WWPT operations.

This use case focuses on the WWTP ASP treatment tanks within its corresponding lanes. It operates in the following manner:

- The wastewater enters the first phase of the primary treatment, where the sediment is clarified.

- The clarified water enters an anaerobic tank, where water gets digested.

- The water slowly moves to the aerobic tank, where the nitrification process happens.

- Finally, the water leaves the tank and other processes are applied, such as second clarification or disinfections.

4.2.1 Data

The data available comprises historical information on three sensors located inside the treatment lane, as shown in **Figure 7**:

- Dissolved Oxygen (DO), located at the entrance of the aerobic tank.

- Water flow, placed at the entrance of the anaerobic tank.

- Ammonia level, set at the final part of the aerobic tank.

These sensors extract the information every 5 minutes, and the dataset spans two years of registers, with regular and irregular values that need to be checked and filtered. Furthermore, due to the sensors being located at different parts of the reaction tank and the water taking time to flow between the inner tanks, it is required to study the time correlation between sensors.

The train set contains 80% of the data, and the test set the remaining 20%. This second set includes the latest data gathered.

4.2.2 Methods

The first phase of the CRISP-DM cycle (Business Understanding) covers the analysis of the problem and the definition of the data-driven approach. The approach focuses on predicting the real-time value of the ammonia parameter using the past and real-time values of the DO and water flow variables, and the past values of ammonia.

Following the CRISP-DM methodology, data are preprocessed, cleaned and new variables are created. To decide which timestamps are used as input features for the model, it is crucial to understand the correlation between them and the objective variable. Pearson's correlation, autocorrelation and cross-correlation techniques [23] are applied to decide the features.

The autocorrelation plot for ammonia is shown on the left graphic in **Figure 8**. The most important lags (previous values) are the ones nearest to the present time, and past hour lags are used as input variables for the model. The cross-correlation among the sensors' data also shows the most important lags. The cross-correlation between the ammonia and water flow variables is shown on the right graph in **Figure 8**, indicating the correlation of any lag from the water flow sensor with the present value of the ammonia sensor. The most important lags are from the previous three hours (−30 lags * 5 minutes per lag), which coincides with the time the water spends moving inside the

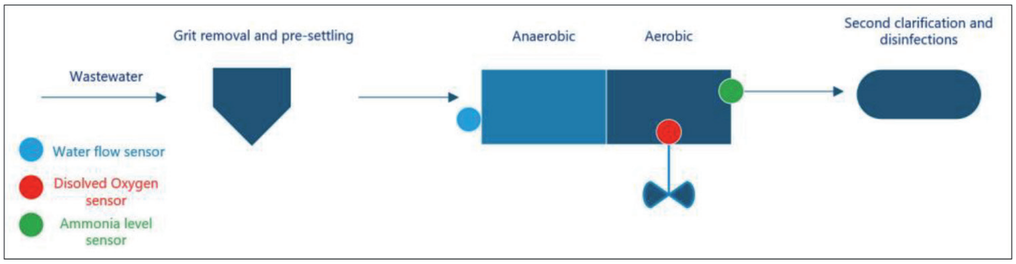

Figure 7.
Wastewater treatment plant lane. Visual sensor location.

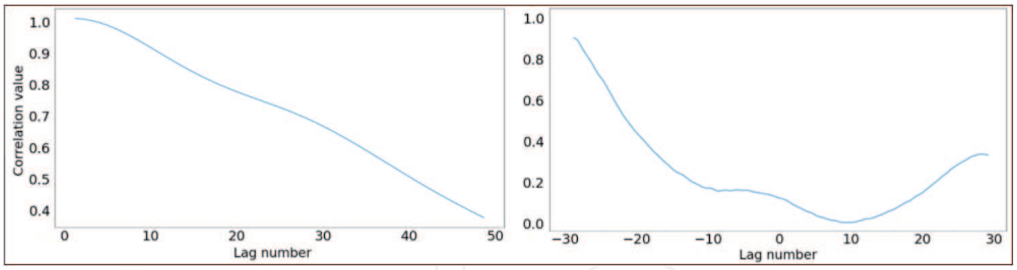

Figure 8.
Left: Autocorrelation for the ammonia parameter. Right: Cross-correlation between the ammonia and water flow parameters.

reaction tank. The DO lag selection follows the same strategy, but in this case, the present values are the most related.

To use the data of the different water parameters, the registers need to have a similar scale of values, so the weight assigned to a feature by the predictive model is not affected by higher or lower values. In this case, the standard score (or Z-score) [24] is used, setting the mean to 0 and scaling the variance to 1.

In the final iterations of the CRISP-DM process, a Long-Short Term Memory (LSTM) [25] Artificial Neural Network [26] algorithm has been used to deal with the process nonlinearities and multiple input time series data, and ultimately, to implement the Ammonia Virtual Sensor. LSTM is a Recurrent Neural Network (RNN) [27] that has feedback connections and can process data sequences such as videos, text, or time series. The inner structure of the LSTM stores the output activations from the different layers of the network. Then, the next time an input is fed, the previously obtained outputs are used as inputs, concatenating the stored information with the new input thus simulating some kind of memory system. The LSTM differentiates from other types of RNNs in the capability of storing multiple iterations of output activations without losing information through time, being the best reason to use this architecture when numerous lags are used. To generate an LSTM architecture, several parameters need to be considered and iterated over. The most important ones are:

1. Number of layers: Number of hidden recurrent layers, to treat the non-linearities of the entering features.

2. Number of neurons: Number of neurons in each layer. Each neuron computes the outputs of the previous layer and sends the result to the next layer.

3. Dropout: Dropout is a regularisation method that probabilistically excludes LSTM units from activating and updating the weight while training, reducing overfitting conditions and therefore improving the model performance. In this use case, the architectures have a dropout of 10%.

To decide the best algorithm hyperparameters (e.g. neural network layer and neurons per layer), several training iterations are done using the Cross-Validation grid search technique over the training dataset to ensure the model is not overfitting. Afterwards, several combinations are compared to find out which combination obtains better results in the test set. The scoring metrics used are the MAE and R^2.

5. Results and discussion

5.1 Aluminium mould injection use case

Using 2 and 5 back samples as additional input variables for the algorithms does not improve the results. Neither the R^2 score nor the MAE score nor the performance improve, but the additional samples hugely increase the prediction time and computational power needed. Therefore, only the same instant sensors' samples, their interactions and the velocity are considered as input variables of the final model.

The predictions of all ML algorithms used in each Virtual Sensor development compared with the real sensor values (black lines) are shown in **Figures 9-11**. For better visual clarity, only four cycles are depicted for each sensor. The prediction and the real values of the first pressure sensor are shown in **Figure 9**. The SVR algorithm (light blue line) and KNN (yellow line) are the algorithms with the lowest R^2 error and highest MAE error for all three sensors. On the contrary, the other two algorithms, Decision Tree (red line) and Random Forest (dark blue line) both present higher R2 errors and lower MAE errors for all three sensors. These metrics can be seen in **Table 1**.

The pressure 2 virtual sensor predictions compared with the real values are shown in **Figure 10**. The results are generally worse in this case than in the pressure 1 sensor. Even though the Random Forest algorithm adjusts more closely to the real sensor, the third and fourth cycle predictions show an example of a fair disparity in the results. It should be kept in mind that the graphic only depicts 4 sample cycles and not the totality of the data predicted. The metrics of the predictions can be found in **Table 2**.

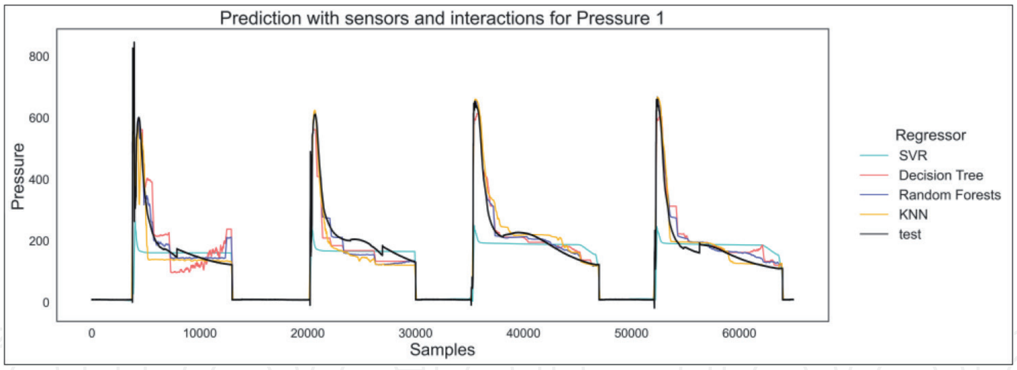

Figure 9.
VS performance comparison for pressure 1 variable simulation.

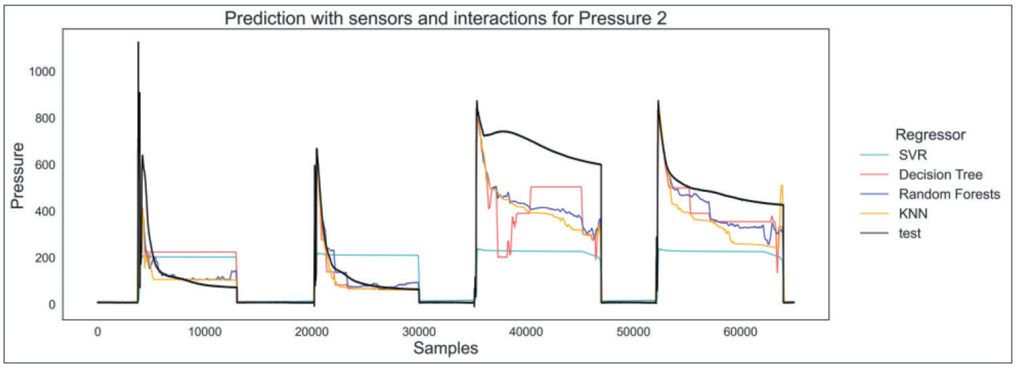

Figure 10.
VS performance comparison for pressure 2 variable simulation.

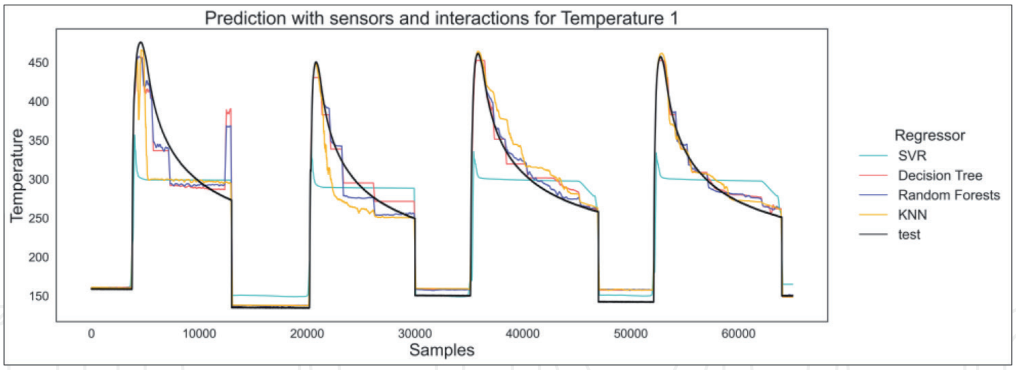

Figure 11.
VS performance comparison for temperature variable simulation.

	Algorithm	Hyperparameters	Train	Validation	Re-train	Test
R2	SVR	C = 1 Epsilon = 0.1	0.709	0.732	—	—
	Decision Tree	Max_depth = 10 Max_features = auto	0.991	0.968	—	—
	Random Forest	N_estimators = 100 Max_ features = 2 Max_depth = 25	0.998	0.972	0.997	0.903
	KNN	N_neighbors = 20 Weights = Distance	0.998	0.717	—	—
MAE	SVR	C = 1 Epsilon = 0.1	35.1	30.2	—	—
	Decision Tree	Max_depth = 10 Max_features = auto	5.19	10.1	—	—
	Random Forest	N_estimators = 100 Max_ features = 2 Max_depth = 25	2.13	9.12	3.12	22.9
	KNN	N_neighbors = 20 Weights = Distance	2.28	23.4	—	—

Table 1.
Performance of each algorithm for the temperature sensor in both datasets.

The results for the temperature sensor are shown in **Figure 11**. In this case also, the SVR predicts almost constant values during the consolidation stage. Unlike the other Virtual Sensor, the prediction during the prefill stage fails to fit closely to the real sensor in all the algorithms.

Tables 1-**3** show the main results in R^2 and MAE for the three Virtual Sensor models and for each algorithm employed. The algorithm with the highest R^2 error and the lowest MAE error is the Random Forest regressor for all three in-mould sensors. Therefore, Random Forests with the mentioned fine-tuned hyperparameters is chosen as the best algorithm. Following the train/test methodology explained beforehand, the performances of the test dataset are also shown in **Tables 1**-**3**.

Both the temperature and pressure 1 sensors obtain high R^2 errors and low MAE errors for the Virtual Sensors predictions. Pressure 2 also gets a high R^2, but high overfitting behaviour can be assumed due to the lower values in the validation and test dataset contrary to the train errors. To illustrate the distribution of the predicted VS values, the counts of the real values versus the predicted ones for the three in-mould

	Algorithm	Hyperparameters	Train	Validation	Re-train	Test
R2	SVR	C = 1 Epsilon = 0.01	0.480	0.461	—	—
	Decision Tree	Max_depth = 10 Max_features = log2	0.966	0.926	—	—
	Random Forest	N_estimators = 120 Max_features = 3 Max_depth = 10	0.998	0.950	0.965	0.820
	KNN	N_neighbors = 20 Weights = Distance	0.997	0.337	—	—
MAE	SVR	C = 1 Epsilon = 0.01	42.1	43.7	—	—
	Decision Tree	Max_depth = 10 Max_features = log2	12.0	20.7	—	—
	Random Forest	N_estimators = 120 Max_features = 3 Max_depth = 10	2.84	16.3	13.7	29.5
	KNN	N_neighbors = 20 Weights = Distance	2.99	44.4	—	—

Table 2.
Performance of each algorithm for the pressure 1 sensor in both datasets.

	Algorithm	Hyperparameters	Train	Validation	Re-train	Test
R2	SVR	C = 0.01 Epsilon = 1	0.381	0.327	—	—
	Decision Tree	Max_depth = 10 Max_features = log2	0.931	0.677	—	—
	Random Forest	N_estimators = 90 max_features = 2 max_depth = 10	0.999	0.775	0.886	0.071
	KNN	N_neighbors = 20 Weights = Distance	0.999	0.482	—	—
MAE	SVR	C = 0.01 Epsilon = 1	64.4	104	—	—
	Decision Tree	Max_depth = 10 Max_features = log2	22.7	57.9	—	—
	Random Forest	N_estimators = 90 max_features = 2 max_depth = 10	2.27	51.0	33.8	97.7
	KNN	N_neighbors = 20 Weights = Distance	2.23	75.9	—	—

Table 3.
Performance of each algorithm for the pressure 2 sensor in both datasets.

sensors are shown in **Figure 12**, using the test dataset. The prediction of pressure 1 is more accurate than pressure 2. In this figure, it can also be observed that although the prediction of pressure 2 is far from making a good prediction, it is worth noting that

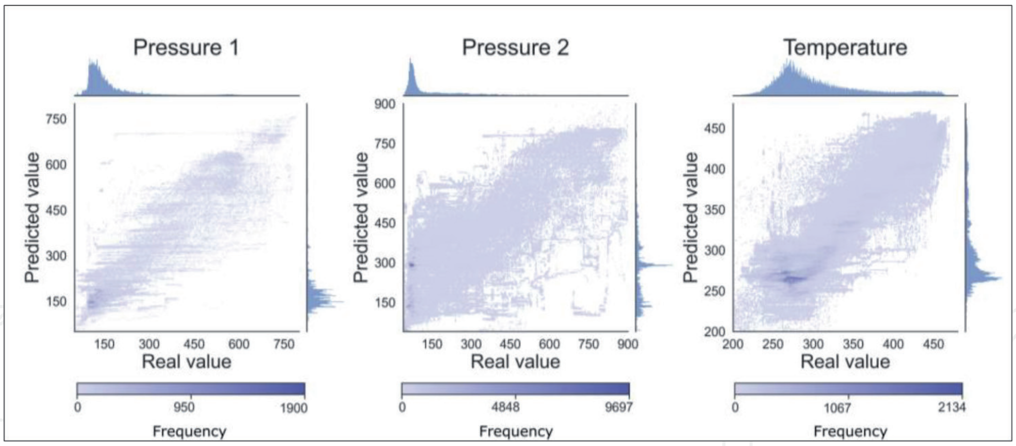

Figure 12.
Heat map of the predicted value vs. real values of each virtual sensor. The colourmap indicates the frequency of repetition.

its errors are mostly due to an erroneous prediction around 0 values, the 'stand-by' value. For the temperature sensor, most values are inside an error of 50 degrees.

5.2 Wastewater treatment plant use case

The train and test processes resulted in the three LSTM architectures outputting the best results are shown in **Table 4**, displaying the scores for the final test set. Similar performances are achieved, but the third model presents the highest R^2. Therefore, the selected architecture is the last one, with 3 hidden layers and 25 neurons on each layer.

The model's response also needs to be validated in situations with a high increase in the ammonia parameter. The Virtual Sensor acting in two cases where the predictions correctly follow the increase of ammonia is shown in **Figure 13**. As it can be seen, the error also increases in these situations since the model is predicting unusual conditions.

To detect possible flaws in the model at a more individual level, the evaluation of registers is done by means of a scatter plot, as shown in **Figure 14**. It compares the predicted and real values, plotting the regression line of all the values to give a general perspective of the overall correlation. It can be observed that, within the predictions, there are no individual registers with a great error, but the general error detected previously is confirmed here. The predictions are lower than the real values, and that is a general flaw of the model trained.

Architectures	MAE	R^2
Number of layers = 4 Number of neurons = 20	0.202	0.960
Number of layers = 3 Number of neurons = 30	0.018	0.970
Number of layers = 3 Number of neurons = 25	0.020	0.975

Table 4.
LSTM architectures and their scoring using the final test set.

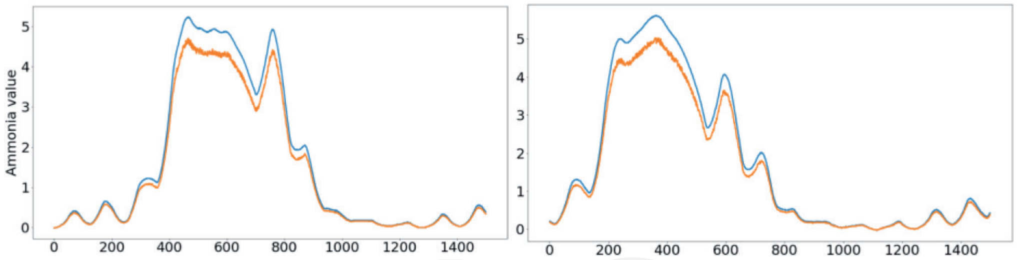

Figure 13.
Real ammonia value, in blue, versus predicted virtual sensor value, in orange. Two cases of a sudden increase in ammonia.

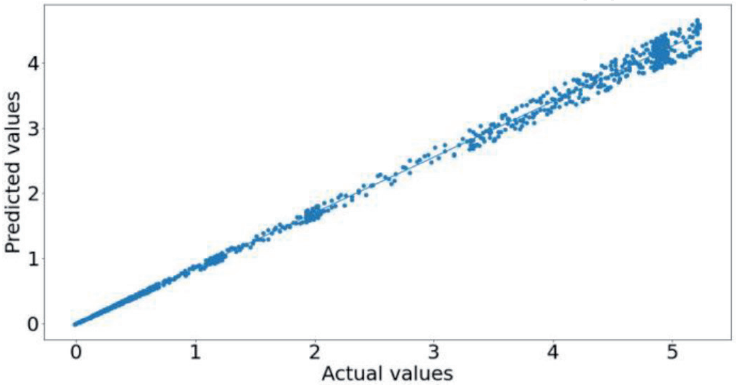

Figure 14.
Scatter plot of the predictions, comparing the individual predictions with the real values.

6. Conclusion

Artificial Intelligence is becoming a key element in the 'must have' technology stack for industries that embrace the challenges and opportunities of the Industry 4.0 paradigm. Smart exploitation of the production chain parameters and data is key for informed decision-making that can impact relevant industrial Key Performance Indicators.

This chapter focuses on a novel approach that utilises Artificial Intelligence and data-driven solutions to expand the production process knowledge base and provide more resilient and robust monitoring systems. The so-called Virtual Sensors allow the creation of indirect measurements of process variables, creating virtual replicas of the real sensors that can detect and mitigate sensors drifts, malfunctions, inaccuracies, etc. Furthermore, new parameters that are difficult or impossible to measure can be estimated by combing inputs of different sensors by means of AI-driven models.

The use of standard methodologies and good practices is considered when describing how the Cross Industry Standard Process for Data Mining can be put in place for developing Virtual Sensor for industrial applications. Additionally, two use cases are presented and described: High Pressure Die Casting (HPDC) and Wastewater Treatment Plant. In the HPDC use case, three Virtual Sensors are implemented to predict two different pressures and the temperature inside the mould cavity. The final models based on Random Forest algorithms offer an R^2 error of 0.903 for the temperature sensors, 0.820 for the pressure 1 sensor and 0.071 for the pressure 2 sensor. The predicted curves follow the real trend, especially for the pressure 1 and temperature sensors, positioning the Virtual Sensors as a trustworthy technology to avoid the implementation of cavity sensors that increase the cost and can affect the shape of the final piece.

In the Wastewater Treatment Plant case, a Virtual Sensors is implemented to improve and ensure the continuous monitoring of the Ammonia parameter in the Activated Sludge Process stage. In this way, the dependence on online real sensor measurements is considerably reduced, which enables an uninterrupted WWTP optimal control. Long-Short Term Memory Deep Neural Network architectures are introduced as algorithms capable to deal with non-linear process behaviours, showing a Deep Learning architecture that correctly adapts to the needs of time series data, which is a good match for Virtual Sensors development. The model benchmarks show a low predictive error, offering a R^2 score of 0.975, thus demonstrating the capacities of such technologies in these complex scenarios.

Acknowledgements

This work was financially supported by the Catalan Government through the funding grant ACCIÓ-Eurecat.

Author details

Maddi Etxegarai*, Marta Camps, Lluís Echeverria, Marc Ribalta, Francesc Bonada and Xavier Domingo
Eurecat, Centre Tecnològic de Catalunya, Unit of Applied Artificial Intelligence, Barcelona, Spain

*Address all correspondence to: maddi.etxegarai@eurecat.org

References

[1] Martin D, Kühl N, Satzger G. Virtual Sensors. Business and Information Systems Engineering. 2021;**63**:315-323. DOI: 10.1007/s12599-021-00689-w

[2] Pech M, Vrchota J, Bednáˇr J. Predictive maintenance and intelligent sensors in smart factory: Review. Sensors. 2021;**21**:1470. DOI: 10.3390/s21041470

[3] Dobrescu R, Merezeanu D, Mocanu S. Process simulation platform for virtual manufacturing systems evaluation. Computers in Industry. 2019;**104**:131-140. DOI: 10.1016/j.compind.2018.09.008

[4] Indri M, Lachello L, Lazzero I, Sibona F, Trapani S. Smart sensors applications for a new paradigm of a production line. Sensors. 2019;**19**(3):650. DOI: 10.3390/s19030650

[5] Maschler B, Ganssloser S, Hablizel A, Weyrich M. Deep learning based soft sensors for industrial machinery. Procedia CIRP. 2021;**99**:662-667. DOI: 10.1016/j.procir.2021.03.115

[6] Alonso S, Morán A, Pérez D, Reguera P, Díaz I, Domínguez M. Virtual sensor based on a deep learning approach for estimating efficiency in chillers. In: International Conference on Engineering Applications of Neural Networks. Cham: Springer; 2019. pp. 307-319. DOI: 10.1007/978-3-030-20257-6_26

[7] Zenisek J, Gröning H, Wild N, Huskic A, Affenzeller M. Machine learning based data stream merging in additive manufacturing. Procedia Computer Science. 2022;**200**:1422-1431. DOI: 10.1016/j.procs.2022.01.343

[8] Ilyas EB, Fischer M, Iggena T, Tönjes R. Virtual sensor creation to replace faulty sensors using automated machine learning techniques. In: 2020 Global Internet of Things Summit (GIoTS). Dublin, Ireland: IEEE; 2020. pp. 1-6. DOI: 10.1109/GIOTS49054.2020.9119681

[9] Tegen A et al. Collaborative sensing with interactive learning using dynamic intelligent virtual sensors. Sensors. 2019;**19**(3):477. DOI: 10.339/s19030477

[10] Djerioui M, Bouamar M, Ladjal M, Zerguine A. Chlorine soft sensor based on extreme learning machine for water quality monitoring. Arabian Journal for Science and Engineering. 2019;**44**:2033-2044. DOI: 10.1007/s13369-018-3253-8

[11] Pattanayak AS, Pattnaik BS, Udgata SK, Panda AK. Development of chemical oxygen on demand (COD) soft sensor using edge intelligence. IEEE Sensors Journal. 2020;**20**:14892-14902. DOI: 10.1109/JSEN.2020.3010134

[12] Foschi J, Turolla A, Manuela A. Soft sensor predictor of E. coli concentration based on conventional monitoring parameters for wastewater disinfection control. Water Research. 2021;**191**:116806. DOI: 10.1016/j.watres.2021.116806

[13] Pisa I, Santín I, Lopez J, Morell A, Vilanova R. ANN-based soft sensor to predict effluent violations in wastewater treatment plants. Water. 2019;**19**(6):1280. DOI: 10.3390/s19061280

[14] Wirth R, Hipp J. CRISP-DM: Towards a standard process model for data mining. In: Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining. Vol. 1. NY, USA: 2000. pp. 29-39

[15] Wolpert DH, Macready WG. No-free-lunch theorems for optimization. IEEE Transactions on Evolutionary Computation. 1995;**1**:67. DOI: 10.1109/4235.585893

[16] Claesen M, De Moor B. Hyperparameter search in machine learning. The XI Metaheuristics International Conference. 2015:1-4. DOI: 10.48550/ARXIV.1502.02127

[17] Benesty J et al. Pearson correlation coefficient. In: Noise Reduction in Speech Processing. Berlin, Heidelberg: Springer; 2009. pp. 1-4. DOI: 10.1007/978-3-642-00296-0_5

[18] Quinlan JR. Induction of decision trees. Machine Learning. 1986;**1**(1):81-106. DOI: 10.1007/BF00116251

[19] Ho TK. Random decision forests. In: Proceedings of 3rd International Conference on Document Analysis and Recognition. Montreal, QC, Canada: IEEE; 1995. pp. 278-282. DOI: 10.1109/ICDAR.1995.598994

[20] Altman NS. An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician. 1992;**46**(3):175-185. DOI: 10.1080/00031305.1992.10475879

[21] Vapnik V. The nature of statistical learning theory. Springer Science & Business Media. 1999. DOI: 10.1007/978-1-4757-2440-0

[22] Metcalf L, Eddy HP, Tchobanoglous G. Wastewater Engineering: Treatment, Disposal, and Reuse. Vol. 4. New York: McGraw-Hill; 1991

[23] Rabiner LR, Gold B, Yuen CK. Theory and application of digital signal processing. IEEE Transactions on Systems, Man, and Cybernetics. Feb 1978;**8**(2):146-146. DOI: 10.1109/TSMC.1978.4309918

[24] Gross E. Practical statistics for High Energy Physics. CERN Yellow Reports: School Proceedings. 2017;**4**:165-186. DOI: 10.23730/CYRSP-2017-004.165

[25] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation. 1997;**9**(8):1735-1780. DOI: 10.1162/neco.1997.9.8.1735

[26] McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics. 1943;**5**(4):115-133. DOI: 10.1007/BF02478259

[27] Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. Parallel Distributed Processing: Explorations in the Microstructure of Cognition. 1986;**1**:318-362. DOI: 10.5555/104279.104293